IN THE SPECIFICATION

In the paragraph beginning on page 9, line 5:

Referring now to FIG. 4, a block diagram of an interconnection of various Stages to form a data flow topology 250 of the underlying forwarding hardware is depicted. However, this data topology is provided as an example, such that those skilled in the art will appreciate that various interconnections of stage objects are within the contemplation of the present invention. The data flow topology 250 includes a link Stage as an input_port 252. An output port 254 of the input_port object 252 is coupled to a classifier Stage, or input classifier 256. The input classifier object 256 is coupled to a scatterer class or demux object 260. The output of the demux object is routed to either a monitor Stage 268 (RMON) or an editor Stage 270 (transcoder object). A gatherer object or gatherer Stage 276 is coupled to the demux object 260 and functions as a layerer object. Finally, a link Stage functions as an output port 280.

In the paragraph beginning on page 10, line 1:

Referring now to FIG. 5, a network 300 utilizing a switching/routing device 302_configured in accordance with the teachings of the present invention is depicted. This network 300 includes the routing device 302 coupled between a wide area network (Internet) 304 and a local area network (private network) 306. The network 300 depicted in FIG. 5 provides the same functionality achieved by the conventional network 100 depicted in FIG. 1. However, rather than using application-specific switching/routing devices, such as utilized in the conventional network 100, the network 300, in accordance with the present invention, utilizes a single switching/routing device to perform each of the packet processing tasks in a single box.

In the paragraph beginning on page 10, line 10:

Referring now to FIG. 6, an internal representation of the switching/routing device 302, as depicted in FIG. 6, is illustrated. The routing device 302 includes a memory 310 containing control plane software 310, and forwarding plane software 320. The device 302 also includes forwarding plane hardware 360. The memory 310 of the router device 302 can be configured to include a compiled and linked-, directed graph of packet processing Stages created using the API model object-oriented software for abstraction, as taught by the present invention. A classification/routing Stage 330 is programmed to perform input processing for the network 300. The directed graph further includes a Stage object 334 configured to perform firewall data path packet processing functionality. The directed graph 330 further includes a Stage object 336 programmed to perform intrusion detection services ID (IDS). Finally, the directed graph 330 includes a Stage object configured to perform virtual private network functionality 338.

IN THE CLAIMS

1. (Currently Amended) A method comprising:

performing identification and decomposition of ~~fundamental units~~operations performed by underlying hardware to process a packet;

forming software objects by abstracting the ~~fundamental units to~~ identification and decomposition of the operations~~packet processing~~ into the software objects, such that the software objects encapsulate and represent functionality performed by underlying hardware to process a packet~~of the fundamental units~~; and

creating an object-oriented programming model using the software objects, such that the software objects ~~modules~~ enable programming of underlying hardware to process packets as programmed using the object-oriented programming model.

2. (Currently Amended) ~~The~~ A method ~~of claim 1, further~~ comprising:

selecting one or more software objects from ~~the~~ an object-oriented programming model;

programming the one or more selected software objects ~~models~~ to perform a desired packet processing functionality; and

connecting the one or more programmed software objects to form a directed graph of packet flow to complete definition of the desired packet processing functionality, such that ~~; and directing~~ underlying hardware ~~loaded with the directed graph of programmed software objects~~ is directed to process packets in accordance with the desired packet processing functionality.

3. (Original) The method of claim 2, wherein the connecting the one or more programmed software objects further comprises:

selecting a first stage object as an input port of the directed graph to direct a physical interface and packet framing;

selecting a second stage object coupled to the first stage object to direct filtering and matching algorithms on packets;

selecting a third stage object coupled to the second stage object to direct routing of packets to one or more third stage object outputs;

selecting a fourth stage object coupled to an output from the one or more third stage object outputs to direct gathering of statistical information regarding packets and packet flows;

selecting an fifth stage object coupled to an output from the one or more third object outputs to direct packet modification;

selecting a sixth stage object coupled to an output from the one or more third object outputs to direct packet routing to a sixth stage object output; and

selecting a final stage object as an output port of the directed graph, the final stage object coupled to the sixth object output.

4. (Currently Amended) The method of claim ~~2~~22, wherein the directing underlying hardware further comprises:

~~making~~ relaying requests from the programmed software objects contained in the direct graph to underlying packet forwarding hardware in accordance with the desired packet processing functionality; and

~~performing packet~~ processing packets by the packet forwarding hardware in response to the software object requests, such that the directed graph of programmed software objects control packet data flow through the packet forwarding hardware.

5. (Original) The method of claim 1, wherein the forming the software objects further comprises:

using as the one or more software objects

a first stage object to define a physical interface and packet framing,

a second stage object to direct filtering and matching algorithms on packets,

a third stage object to direct packet flow policy,

a fourth stage object to direct packet routing to scatterer object outputs,

a fifth stage object to direct packet collecting and routing scattered packets;

an sixth stage object to direct packet modification,

a seventh stage object to direct packet flow policy, and

an eighth stage object to direct gathering of statistical information regarding packets and packet flows.

6. (Currently Amended) The method of claim 5, wherein the one or more software objects each include one or more inputs and one or more outputs enabling formation of compositions of objects sharing a common interface to direct packet~~s~~ processing as a group of software objects and to form~~ing~~ directed graphs of software objects to direct packet data flow through ~~the~~ packet forwarding hardware.

7. (Currently Amended) A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, cause the processor to performing a method comprising:

~~performing identification and decomposition of fundamental units performed by underlying hardware to process a packet;~~

~~forming software objects by abstracting the fundamental units to packet processing into the software objects, such that the software objects encapsulate and represent functionality of the fundamental units; and~~

~~creating an object-oriented programming model using the software objects, such that the software object modules enable programming of underlying hardware to process packets as programmed using the object-oriented programming model.~~

directing underlying hardware loaded with a directed graph of programmed software objects to process packets in a manner specified using an object-oriented model and compiled to produce the directed graph of programmed software objects.

8. (Cancelled)

9. (Cancelled)

10. (Currently Amended) The machine-readable medium of claim 8, wherein the directing underlying hardware further comprises:

~~making~~ relaying requests from the software objects contained in the direct graph to underlying packet forwarding hardware in accordance with ~~the~~ a desired packet processing functionality; and

performing packet processing by the packet forwarding hardware in response to the software object requests, such that the directed graph of programmed software objects control packet data flow through the packet forwarding hardware.

11. (Cancelled)

12. (Cancelled)

13. (Currently Amended) An apparatus, comprising:

a processor; and

a memory coupled to the processor, the memory to load a directed graph of programmed software objects ~~and including software object linked together to from a directed graph of packet flow~~ to direct the processor to ~~perform packet processing~~ process packets in a manner specified using an object-oriented ~~programming~~ model and compiled to produce~~d~~ the directed graph of programmed software objects.

14. (Currently Amended) The apparatus of claim 13, wherein the memory is further ~~configured to~~ relay requests from the software objects contained in the direct graph to the processor, such that processor ~~performs~~ processes packets ~~processing~~ in response to the software object requests ~~for~~ to control packet data flow through the processor.

15. (Currently Amended) The apparatus of claim 13, wherein the processor ~~is configured as~~ comprises a network processor.

16. (Currently Amended) The apparatus of claim 13, wherein the processor ~~is configured as a~~ comprises an application specific integrated circuit.

17. (Currently Amended) A system comprising:

a wide area network;

a local area network; and

a processor coupled between the wide area network and the local area network on to form a network, the processor having

a memory coupled to the processor, the memory to load a directed graph of programmed software objects ~~and including software objects linked together to from a directed graph of packet flow~~ to direct the processor to ~~perform~~ process packets ~~processing~~ in a manner specified using an object-oriented ~~programming~~ model and compiled to produced the directed graph of programmed software objects.

18. (Currently Amended) The system of claim 17, wherein the memory is further configured to relay requests from the software objects contained in the directed graph to the processor, such that processor ~~performs~~ process packets ~~processing~~ in response to the software object requests for controlling packet data flow through the processor.

19. (Currently Amended) The system of claim 17, wherein the processor ~~is configured as~~ comprises a network processor.

20. (Currently Amended) The system of claim 17, wherein the processor ~~is configured as~~ comprises an application specific integrated circuit.

21. (Currently Amended) The system of claim 17, wherein each software object within the directed graph performs a data-path packet processing task functionality, such that the directed

graph of programmed software objects performs a plurality of data-path packet procession tasks ~~with~~ within a single device.

Please add the following new claims:

22. (New) The method of claim 2, further comprising:

loading underlying hardware with the directed graph of programmed software objects to process packets in accordance with the desired packet processing functionality.

23. (New) The method of claim 4:

wherein each software object within the directed graph performs a data-path packet processing task functionality, such that the directed graph of programmed software objects performs a plurality of data-path packet processing tasks within a single device.

24. (New) The machine readable medium of claim 10,

wherein each software object within the directed graph performs a data-path packet processing task functionality, such that the directed graph of programmed software objects performs a plurality of data-path packet processing tasks within a single device.

25. (New) A method comprising:

directing underlying hardware loaded with a directed graph of programmed software objects to process packets in a manner specified using an object-oriented model and compiled to produce the directed graph of programmed software objects.

26. (New) The method of claim 25, wherein the directing underlying hardware further comprises:

relaying requests from the programmed software objects contained in the direct graph to underlying packet forwarding hardware in accordance with the desired packet processing functionality; and

processing packets by the packet forwarding hardware in response to the software object requests, such that the directed graph of programmed software objects control packet data flow through the packet forwarding hardware.

27. (New) The method of claim 26, wherein each software object within the directed graph performs a data-path packet processing task functionality, such that the directed graph of programmed software objects performs a plurality of data-path packet processing tasks within a single device.